UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/735,217	12/12/2003	Vishal Saxena	ORA011 US	1761

56135	7590	06/20/2006

SILICON VALLEY PATENT GROUP LLP
2350 MISSION COLLEGE BLVD.
SUITE 360
SANTA CLARA, CA 95054

EXAMINER
OPARE ABETIA, JOSEPH C

ART UNIT	PAPER NUMBER
2165	

DATE MAILED: 06/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/735,217	SAXENA, VISHAL
	Examiner	Art Unit	
	Joseph C. Opare-Abetia	2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _12 December 2003_.

2a)☐ This action is **FINAL.**   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle,* 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-26_ is/are pending in the application.

　　4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-26_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

　　Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

　　Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

　　a)☐ All   b)☐ Some * c)☐ None of:

　　　1.☐ Certified copies of the priority documents have been received.

　　　2.☐ Certified copies of the priority documents have been received in Application No. _____.

　　　3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

　　* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _01/11/2005_.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) The invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-26 are rejected under 35 U.S.C. 102(b) as being anticipated by Peter

Rex Thomas et al. (U.S. Patent No. 6460052 and Thomas hereinafter).

With respect to claim 1, Thomas discloses a method of managing a repository

containing multiple versions of an object, the method comprising: inserting into a first

table a first row comprising at least one attribute of a first object, an identifier of the first

object, and a version number of the first object (i.e., *"As previously indicated, the version*

control mechanism provides an object identity scheme that associates additional identity

information with every object version within the repository. FIG. 2 illustrates a repository 200

that includes a table of objects 208, a list of configuration members 206, a working context table

204, a user workspace 202 and a set of tools 240. Object table 208 includes a plurality of rows

or entries that are each associated with a specific version of a particular object within repository

200." The preceding text clearly indicates that each object inserted has identification and a

version number.)(Col.6 lines 37-45); and inserting into a second table a second row

comprising the version number of the first object, the identifier of the first object, an

identity of a configuration wherein the first object exists, and at least one identifier of a

identity of a configuration wherein the first object exists, and at least one identifier of a

second object to which said first object is related in said configuration (i.e., "...*Each table*

definition in the second set of table definitions includes columns that correspond to the columns

of the corresponding table definition in the first set of table definitions and one or more

additional columns for storing version information." The preceding text clearly indicates that the

second set of table is related to that of the first set of table)(col. 2 lines 64-67, col. 3 lines 1-5).

With respect to claim 2, Thomas discloses a method further comprising: receiving

a query for the first object and for the second object related to the first object in said

configuration (i.e., "*Each table definition in the second set of table definitions includes columns*

that correspond to the columns of the corresponding table definition in the first set of table

definitions." The preceding text clearly indicates that the second object is related to the first

object)(abstract); using the configuration to retrieve from the second table: the version

number, the identity of the first object, and the identity of the second object (i.e., "*In*

response to a request from a user to retrieve the particular object, a version of the particular

object to present to the user is determined based on a workspace associated with the user." The

preceding text clearly indicates that retrieval of elements such as version number and

identification is being implemented in the system)(col. 3 lines 12-18); using the version

number and the identity of the first object to retrieve from the first table: the attribute of

the first object (See explanation above) (col. 3 lines 12-18); using the configuration and the

identity of the second object to retrieve from a third table: the version number of the

second object (see explanation above)(col. 3 lines 12-18); and using the version number of

the second object and the identity of the second object to retrieve from a fourth table: an

attribute of the second object (see explanation above)(col. 3 lines 12-18).

With respect to claim 3, Thomas discloses a method further comprising: checking

if the first object has a parent object and if a copy of the parent object exists in a

previously stored configuration and if both conditions are true, inserting a third row for

the parent object in a third table using the identifier of the parent object, and an

incremented version number of the parent object, and inserting a fourth row into a fourth

table using the incremented version number of the parent object, the identifier of the

parent object, and an identity of the configuration wherein the first object exists (i.e.,

"...*the version control mechanism inserts a default genealogy ID and unique version ID value in*

the genealogy ID column and unique version ID column for each table entry." The preceding

text clearly indicates that the system inserts columns and row for ID's corresponding to related

objects)(col. 9 lines 5-9).

With respect to claim 4, Thomas discloses a method wherein the second table

has a column for a minimum version number of the parent object and another column

for a maximum version number of the parent object, and the method further comprises:

if both of said conditions are true, storing in the second row the incremented version

number of the parent object as the minimum version number (i.e., "*In one embodiment, in*

registering a schema, the repository metadata is updated or "populated" to include the newly

generated version enabled schema information." The preceding text clearly indicates that the

system update by adding new version to the system)(col. 9 lines 32-35); if a copy of the parent

object does not exist in a previous configuration then storing in the second row the

version number of the parent object as the minimum version number (i.e., *"When*

information objects are stored in a relational database that does not have a version control

mechanism, they are generally stored in rows of one or more tables, where the tables have

columns for storing the values of the various attributes of the objects.")(col. 4 lines 44-48); and

setting the maximum version number as infinity (the elements on figure 2 clearly indicates

objects being stored as minimum and maximum as being infinity illustrated by the doted

elements below the table)(Fig. 2).

With respect to claim 5, Thomas disclose a method further comprising receiving

an instruction to update the first object, and checking if a copy of the first object exists in

a previous configuration and if not updating the object (i.e., *"when an object is inserted,*

deleted or updated within the view 350, the triggers evaluate the access control to determine

whether or not the operation can be performed." The preceding text clearly indicates that the

system is performing updating which can also be claim as checking if a copy of the first object

exists in a previous configuration)(col.11 line 40-43).

With respect to claim 6, Thomas discloses a method wherein if a copy of the first

object exists in a previous configuration, the method further checking if the first object

has a parent object and if not, inserting a new row in the first table for the first object

using an incremented version number and inserting a new row in the second table for

relationships of the first object (i.e., "...*the version control mechanism inserts a default genealogy ID and unique version ID value in the genealogy ID column and unique version ID column for each table entry.*" The preceding text clearly indicates that the system inserts columns and row for ID's corresponding to related objects)(col. 9 lines 5-9).

With respect to claim 7, Thomas discloses a method further comprising checking if a copy of the parent object exists in a previous configuration and if true: inserting a third row for the parent object in a third table using an identifier of the parent object and an incremented version number of the parent object (i.e., "*In one embodiment, in registering a schema, the repository metadata is updated or "populated" to include the newly generated version enabled schema information*". The preceding text clearly indicate that the system updates when newly generated version are created which can also be said as adding rows and columns for the original object which in this application, the applicant calls it parent object)(col. 9 lines 32-34); inserting a fourth row into a fourth table using the incremented version number of the parent object and the identifier of the parent object (see explanation above); setting in the second table a maximum version number of the parent object to be the version number of the parent object prior to incrementing (see explanation above); inserting into the first table a fifth row comprising said attribute of the first object, the identifier of the first object, and an incremented version number of the first object (i.e., "*The method includes creating a table that includes a first set of one or more columns for storing values that correspond to attributes of objects, and a second set of one or more columns for storing object version information. Multiple versions of a particular object*

are stored within the table.")(col. 3 lines 8-12); inserting into the second table a sixth row

comprising the incremented version number of the first object, the identifier of the first

object, the incremented version number of the parent object as a minimum number of

the parent object and infinity as the maximum version number of the parent object (the

elements on figure 2 clearly indicates objects being stored as minimum and maximum as being

infinity illustrated by the doted elements below the table)(Fig. 2).

With respect to claim 8, Thomas discloses a method of further comprising

receiving an instruction to delete the first object, and checking if a copy of the first object

exists in a previous configuration (i.e., "*If the operation is allowed, the triggers construct the*

subordinate SQL that is necessary to insert, delete or update the object within the TABLE $A and

create the configuration members so that the object can be seen in a subsequent select operation

back through the view 350, inserts the object into the current working folder so that it will be

seen when navigating the folder, and does the background evaluation of the constraints for

generating the corresponding version history information." The preceding text clearly indicates

that implement the phase of updating file versions in the system)(col. 11 lines 43-52).

With respect to claim 9, Thomas discloses a method wherein if the result of

checking is not true, the method further comprising: deleting the first row and the

second row from the first table and the second table respectively (i.e., "*If the operation is*

allowed, the triggers construct the subordinate SQL that is necessary to insert, delete or update

the object within the TABLE $A and create the configuration members so that the object can be

seen in a subsequent select operation back through the view 350, inserts the object into the

current working folder so that it will be seen when navigating the folder, and does the

background evaluation of the constraints for generating the corresponding version history

information." The preceding text clearly indicates that implement the phase of updating file

versions in the system by deleting and inserting files)(col. 11 lines 43-52).

With respect to claim 10, Thomas discloses a method further comprising:

checking if the first object has a child object and if so whether a minimum version

number of said child object in a third table is same as the version number of the first

object in the first table and if so, deleting a row for the child object from the third table

and deleting another row for the child object from a fourth table (i.e., " *inserts the object*

into the current working folder so that it will be seen when navigating the folder, and does the

background evaluation of the constraints for generating the corresponding version history

information." The preceding text clearly indicates that implement the phase of updating file

versions in the system by deleting and inserting files)(col. 11 lines 43-52).

With respect to claim 11, Thomas discloses a method wherein if the result of

checking is true, the method further comprising: deleting the second row from the

second table (i.e., "*If the operation is allowed, the triggers construct the subordinate SQL that*

is necessary to insert, delete or update the object within the TABLE $A and create the

configuration members so that the object can be seen in a subsequent select operation back

through the view 350, inserts the object into the current working folder so that it will be seen

when navigating the folder, and does the background evaluation of the constraints for generating the corresponding version history information." The preceding text clearly indicates that implement the phase of updating file versions in the system by deleting and inserting files)(col. 11 lines 43-52); and checking if the first object has a parent and if true inserting a third row for the parent object in a third table using an identifier of the parent object and an incremented version number of the parent object, inserting a fourth row into a fourth table using the incremented version number of the parent object and the identifier of the parent object, and setting in the second table a maximum version number of the parent object to be the version number of the parent object prior to incrementing (See explanation above).

With respect to claims 12, 14, 23,and 25 Thomas discloses computer-readable storage medium encoded with a sequence of instructions to perform (i.e., *"Execution of the sequences of instructions contained in main memory 606 causes processor 604 to perform the process steps described herein.*" The preceding text clearly indicates that a signal is sent to a sequence of instructions to be executed)(col. 15 lines 20-23).

With respect to claim 13, Thomas discloses a storage medium further comprising the repository, the second table comprising a first column for holding a maximum version number of a parent of the first object and a second column for holding a minimum version number of the parent of the first object, a current version number of the parent being stored in the second column (The elements on figure 2 clearly indicates

objects being stored as minimum and maximum as being infinity illustrated as doted below the

table)(fig. 2).

With respect to claim 15, Thomas discloses a computer comprising: a storage

medium comprising a repository containing multiple versions of a first object (i.e.,

"*repository is a storage mechanism... repository stores and versions both relational data and file*

based data... The repository maintains version history information about the different objects

that it stores..." The preceding text clearly indicates that the repository stores information

relating to object such as versions)(col. 5 line 58, col. 6 lines 1-24); means, coupled to the

storage medium, for inserting into a first table in the repository a first row comprising at

least one attribute of the first object, an identifier of the first object, and a version

number of the first object (i.e., "*As previously indicated, the version control mechanism*

provides an object identity scheme that associates additional identity information with every

object version within the repository. FIG. 2 illustrates a repository 200 that includes a table of

objects 208, a list of configuration members 206, a working context table 204, a user workspace

202 and a set of tools 240. Object table 208 includes a plurality of rows or entries that are each

associated with a specific version of a particular object within repository 200." The preceding

text clearly indicates that each object inserted has identification and a version number.)(Col.6

lines 37-45); and means, coupled to the storage medium, for inserting into a second

table in the repository a second row comprising the version number of the first object,

the identifier of the first object, an identity of a configuration wherein the first object

exists, and an identifier of a second object to which said first object is related (i.e.,

"*...Each table definition in the second set of table definitions includes columns that correspond to the columns of the corresponding table definition in the first set of table definitions and one or more additional columns for storing version information.*" The preceding text clearly indicates that the second set of table is related to that of the first set of table)(col. 2 lines 64-67, col. 3 lines 1-5).

With respect to claim 16, Thomas discloses a method of managing a repository containing multiple versions of an object, the method comprising: inserting into a first table, a row for a first object being added to the repository, the first table comprising a column for a minimum version number of a second object and another column for a maximum version number of the second object (figure 3 clearly shows that the system implementing the method of inserting version numbers in columns and rows)(fig. 3); and in response to a query to which a plurality of versions of the first object are responsive, selecting a version of the first object for which a version of the second object falls between the minimum version number and the maximum version number, wherein said version of the second object is responsive to the query (i.e., "*In response to a request from a user to retrieve the particular object, a version of the particular object to present to the user is determined based on a workspace associated with the user. The version of the particular object is presented to the user without exposing values from the second set of one or more columns to the user*" the preceding text clearly indicates that upon sending a query the requester receives some versions pertaining to that particular object without exposing all of the version associated with that object)(abstract).

With respect to claim 17, Thomas discloses a method of wherein the second object contains the first object either directly or indirectly (i.e., *"Each table definition in the second set of table definitions includes columns that correspond to the columns of the corresponding table definition in the first set of table definitions."* The preceding text clearly indicates that the second object is contained in the first object)(abstract).

With respect to claim 18, Thomas discloses a method wherein the second object is an immediate parent of the first object (i.e., *"The method includes reading a first set of table definitions that belong to the non-versioned schema. A second set of table definitions is generated for the version-enabled schema such that each table definition in the second set of table definitions corresponds to a table definition in the first set of table."* The preceding text clearly indicates that second object was generated from the first so one can claim of second object is an immediate parent of the first object or vice versa)(abstract)

With respect to claim 19, Thomas discloses a method wherein the first table is devoid of any information that defines the first object (i.e., *"After the rows that do not correspond to object versions that are in the user's working contexts have been filtered, the version control mechanism strips off any added object identity information (genealogy ID and unique version ID)."* The preceding text clearly indicates that or information does not fit within the file-based therefore one skilled in the art can claim of first table is devoid of any information that defines the first object)(col. 10 lines 8-12).

With respect to claim 20, Thomas discloses a method wherein the repository comprises a second table, the second table comprises a column for an identity of a configuration to which the second object belongs, and the first table is devoid of said identity of configuration (i.e., *"method includes creating a table that includes a first set of one or more columns for storing values that correspond to attributes of objects, and a second set of one or more columns for storing object version information After the rows that do not correspond to object versions that are in the user's working contexts have been filtered, the version control mechanism strips off any added object identity information (genealogy ID and unique version ID) and projects the data to the existing tools. Thus, the existing tool set can continue to use its normal SQL queries to access the data as if it still existed in the old tables."* The preceding text clearly indicates that element such as object identifier correspond to objects in the table)(col. 3 lines 8-11, col. 10 lines 8-15)

With respect to claim 21, Thomas discloses a method wherein the first table comprises an additional column for an identity of a configuration to which the first object belongs, and the first table further comprises yet another column for an attribute that at least partially defines the first object (i.e., *"When information objects are stored in a relational database that does not have a version control mechanism, they are generally stored in rows of one or more tables, where the tables have columns for storing the values of the various attributes of the objects."* The preceding text clearly indicates that more columns are added to store objects information)(col. 4 lines 44-48).

With respect to 22, Thomas discloses a method wherein the second object is not

contained in any other object (i.e., *"alternatively, because entries 214 and 218 do not have the*

same genealogy ID, the object versions associated with entries 214 and 218 are identified as

being from different families." The preceding text clearly indicates how objects are different than

each other)(col. 6lines 54-57)

With respect to claim 24, Thomas discloses a storage medium further comprising

the repository, and a second table comprising a pair of columns defining a range of

version numbers of a parent of the first object, wherein a current version number of the

parent falls within the range (i.e., *"...a configuration itself may actually include other*

configurations. For example, as depicted by entry 234, the configuration associated with version

"1" of configuration ID "1" also includes the objects that are contained in version "1" of

configuration Working context table 204 defines the list of configuration objects that are

currently mapped to a user's workspace." The preceding texts clearly indicates that a range is set

for every object in the database therefore, one skilled in the art can claim of a columns defining a

range of version numbers of a parent of the first object, wherein a current version number of the

parent falls within the range)(col. 7 lines 45-51).

With respect to claim 26, Thomas discloses a storage medium comprising a

repository containing multiple versions of a first object (i.e., *"Multiple versions of a*

particular object are stored within the table." The preceding text clearly indicates that multiple

version of an object is stored in a database therefore one can give the it a name as first

object)(Abstract); means, coupled to the storage medium, for inserting into a first table in

the repository a first row for a first object to be added to the repository, the first table

comprising a column for a minimum version number of a second object and another

column for a maximum version number of the second object, wherein a version number

of the second object is stored as the minimum version number, and infinity is stored as

the maximum version number (the elements on figure 2 clearly indicates objects being stored

as minimum and maximum as being infinity illustrated as doted below the table)(Fig. 2); and

means, coupled to the storage medium, for selecting a version of the first object for

which a version of the second object falls between the minimum version number and

the maximum version number, wherein said version of the second object is responsive

to a first portion of a query, and a plurality of versions of the first object including said

selected version are responsive to a second portion of said query (i.e., "*In response to a

request from a user to retrieve the particular object, a version of the particular object to present

to the user is determined based on a workspace associated with the user. The version of the

particular object is presented to the user without exposing values from the second set of one or

more columns to the user*" The preceding text clearly indicates that upon sending a query the

requester receives some versions pertaining to that particular object without exposing all of the

version associated with that object. The elements on figure 2 clearly indicates objects being

stored as minimum and maximum as being infinity illustrated as doted below the table)(abstract;

fig. 2).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joseph C. Opare-Abetia whose telephone number is (571) 272-6594. The examiner can normally be reached on mon-fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JEFFREY A. GAFFIN can be reached on (571) 272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

FRANTZ COBY
PRIMARY EXAMINER

Joseph C. Opare-Abetia

Patent Examiner

Art Unit 2165